MISB MOTION IMAGERY STANDARDS BOARD

MISB Standard 0604.1

STANDARD

30 September 2010

Time Stamping and Transport of Compressed Motion Imagery and Metadata

1 Scope

This STANDARD defines methods to time stamp compressed video streams and to transport video and metadata asynchronously or synchronously in compressed motion imagery streams.

Implementation methods are defined that leverage the transport layer of MPEG-2 for carriage of motion imagery streams of varying types and bit rates as defined in the Motion Imagery Standards Profile concept of "Xon2". Specific compressed video formats covered include MPEG-2 and H.264.

2 References

[1] ISO/IEC 13818-1:2007 Information technology - *Generic coding of moving pictures and associated audio information: Systems*

[2] ISO/IEC 13818-2:2000 Information technology - *Generic coding of moving pictures and associated audio information: Video*

[3] ISO/IEC 14496-10:2008 Information technology - *Coding of audio-visual objects - Part 10: Advanced Video Coding*

[4] MISB RP 0101, *Use of MPEG-2 System Streams in Digital Motion Imagery Systems*

[5] MISB RP 0603, *Common Time Reference for Digital Motion Imagery using Coordinated Universal Time (UTC)*

[6] MISB STANDARD 9715, *Time Reference Synchronization*

[7] Motion Imagery Standards Profile 5.0, *DoD/IC/NSG Motion Imagery Standards Board*, http://www.ismc.nga.mil

[8] SMPTE 12M-1999, Television, Audio and Film - *Time and Control Code*

[9] SMPTE RP 188-1999, *Transmission of Time Code and Control Code in the Ancillary Data Space of a Digital Television Data Stream*

[10] SMPTE 328M-2000, *MPEG-2 Video Elementary Stream Editing Information*

[11] SMPTE RP 217-2001, *Nonsynchronized Mapping of KLV Packets into MPEG-2 System Streams*

[12] SMPTE 309M-1999, *Transmission of Date and Time Zone Information in Binary Groups of Time and Control Code*

[13] SMPTE 336M-2007, *Data Encoding Protocol using Key-Length-Value*

3 Introduction

The MPEG-2 transport layer [1] provides an infrastructure for the carriage of video, audio and metadata in a single motion imagery stream as shown in the following diagram.

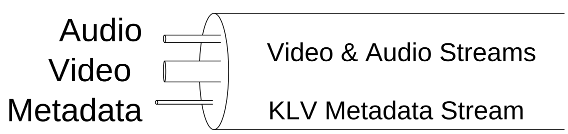

Figure 1: MPEG-2 Transport Stream

The Motion Imagery Standards Profile (MISP) [7] endorses the use of MPEG-2 Transport Streams for this purpose. The MISB has been researching the use of MPEG-2 Transport Streams for the carriage of other motion imagery formats in a study known as "Xon2". Recent recommendations extend the use of MPEG-2 Transport Streams as a means for carriage of H.264 video in the compressed domain as defined in [1].

The advantages of using Universal Coordinated Time (UTC) as the master clock reference for video and metadata are outlined in [5], which discusses several time formats and the relationships between them. This STANDARD defines how the UTC time can be used to stamp MPEG-2 and H.264 video streams, and how the video and metadata can be synchronously transported in motion imagery streams.

Motion imagery analysis and processing applications require various levels of temporal accuracy when referencing metadata elements and the video frames associated with those elements. Compressed imagery generated from standard definition analog video sensors has traditionally utilized asynchronous methods for carriage of metadata in a private data stream. This was adequate for metadata that was not time sensitive, or for metadata which only needed to be associated to within a few seconds of the correct video frame. Asynchronous transport could not be used reliably for systems which required metadata to be frame or sub-frame accurate.

Synchronous multiplexing of metadata with video ensures that the proximity between a metadata item and the associated video is well defined. This in turn reduces the latency in the system and helps prevent the metadata from being separated from the associated video when the video is processed.

This STANDARD provides guidance on methods to synchronously transport video frames and associated metadata elements with varying levels of precision as determined by the user's requirements.

4 Time Stamping Video

System designers should be aware of the accuracy requirements for the time stamps in their system. The use of UTC as a deterministic common time reference for the correlation of motion imagery frames and metadata is defined in [5], which also describes several types of systems and the relative accuracies of each.

Time stamps may be introduced into a compressed video stream in one of two ways. If the uncompressed video signal contains a time stamp in the Vertical Interval Time Code (VITC) or the Vertical Ancillary Data Space (VANC), it is recommended that the video encoder extract the time stamp from the VITC or VANC of the incoming video signal and insert it into the video elementary stream as indicated in the following sections.

If the uncompressed video signal does not contain a time stamp, the encoder should be enabled to read the time stamp from the system time clock or an external source and insert it into the video elementary stream.

The following sections describe how to insert the time stamp into MPEG-2 and H.264 video streams.

4.1 MPEG-2

Standard 9715 [6] "Time Reference Synchronization" states that:

> Universal coordinated time (UTC, also known as "Zulu"), clock signals shall be used as the universal time reference for DoD/IC/NSG SMPTE 12M time code systems, allowing systems using time code to accurately depict the actual Zulu time of day of motion imagery acquisition / collection / operations.

The following sections describe how to use the Group of Pictures (GOP) time code to time stamp MPEG-2 compressed video, and how a time stamp in the video elementary stream user data field or a time stamp in the MPEG-2 video elementary stream editing information may be used in systems which require a more persistent time stamp or one with a higher level of precision.

4.1.1 GOP Time Code

The MPEG-2 video layer includes the definition of a time code within the GOP header. This time code is of the form HH:MM:SS:FF in a format specified in [8].

It is strongly recommended that the SMPTE time code in the GOP header be filled in with a time stamp which represents UTC time for MPEG-2 video streams for all motion imagery systems.

The accuracy of the SMPTE 12M time code as it is inserted into the video signal for systems with integer and non-integer frame rates is indicated in [5], and for cameras which are or are not phase locked to the master time reference.

For systems which process signals with integer frame rates, and for video sources that are genlocked to a UTC time reference, the accuracy of the time stamp in the GOP header can be quite accurate (sub-frame accuracy). The accuracy decreases for systems with non-integer frame rates.

The usefulness of the GOP time code has some limitations:

- The GOP time code is generally not persistent, and not considered by the MPEG-2 standards to be an absolute time. When a video is edited, the editor will often re-stamp the GOP time code.
- The GOP time code includes a time, but not a date. The date information, if needed, must be extracted from the KLV metadata in the stream.
- The accuracy of the GOP time code is limited, particularly in motion imagery with non-integer frame rates.

Some of these limitations can be addressed by also populating a time code in the elementary stream user data or MPEG-2 video elementary stream editing information as described in the following sections.

4.1.2 Elementary Stream User Data

The MPEG-2 format allows user defined data to be inserted into the video elementary stream in a user data field (start code value = 0xB2). The video specification [2] allows the user data field to be placed in several different places in the video bitstream. The user data field containing the time stamp is placed between the picture header and the picture data so that it relates to a frame of video.

The elementary stream user data time stamp may be used in systems which are required to associate a highly accurate, microsecond resolution time stamp with the video frame. This UTC time stamp shall be derived from GPS as described in section 4 of [5] and will be formatted as defined in Annex A of [5]. The user data message consists of an identification string and a time stamp as defined below:

Identification String: 16 Bytes that shall be set to the value:

Bytes 1-8:	0x4D, 0x49, 0x53, 0x50, 0x6D, 0x69, 0x63, 0x72,
Bytes 9-16:	0x6F, 0x73, 0x65, 0x63, 0x74, 0x69, 0x6D, 0x65

This represents the ASCII string: “MISPmicrosectime”

Time Stamp: 12 additional bytes defined as follows:

Byte 17:		Status
Bit 7	0 =	GPS Locked (internal clock locked to GPS)
	1 =	GPS Flywheel (internal clock not locked to GPS, so it is running on an internal oscillator)
Bit 6	0 =	Normal (time incremented normally since last message)
	1 =	Discontinuity (time has not incremented normally since last message)
Bit 5	0 =	Forward (If Bit 6 = 1, this indicates that the time jumped forward)
	1 =	Reverse (If Bit 6 = 1, this indicates that the time jumped backwards)
Bits 4-0:		Reserved (= 1)
Bytes 18, 19:		Two MS bytes of Time Stamp (microseconds)
Byte 20:		Start Code Emulation Prevention Byte (0xFF)
Bytes 21, 22:		Two next MS bytes of Time Stamp (microseconds)
Byte 23:		Start Code Emulation Prevention Byte (0xFF)
Bytes 24, 25:		Two next LS bytes of Time Stamp (microseconds)
Byte 26:		Start Code Emulation Prevention Byte (0xFF)
Bytes 27, 28:		Two LS bytes of Time Stamp (microseconds)

This represents the 64 bit microsecond UTC time stamp where Byte 18=MSB, Bytes 19, 21, 22, 24, 25, 27 are intermediate bytes and Byte 28 = LSB. Byte 1 is transmitted first.

4.1.3 MPEG-2 Video Elementary Stream Editing Information

Additional information that may be carried in the user data area of a video elementary stream is described in [10]. One of the additional metadata elements is a 64 bit time code which complies with [8] and [12]. The time code represents the time that the frame was captured (HH:MM:SS:FF), and it contains a date as defined in [12].

4.2 H.264

As with MPEG-2, the H.264 compression format provides places to include a time stamp in the video stream. Both of the time stamps described below are placed in the Supplemental Enhancement Information (SEI) message.

4.2.1 Pic_Timing Time Stamp

The H.264 format, specified in [3] provides for an optional time stamp to be defined in the SEI Message. The "picture timing SEI message" (pic_timing) specifies the time as HH:MM:SS:FF. It is a persistent time stamp which reflects the time of frame capture and it contains flags to specify whether the video is drop-frame, and whether there is a discontinuity in the video time-line.

For H.264 compressed motion imagery, it is strongly recommended that the pic_timing field in the SEI Message be filled in with a time stamp which represents UTC time for H.264 video streams for all motion imagery systems.

4.2.2 User Data

The H.264 format also allows user defined data to be associated with a particular video frame using the user data unregistered SEI Message.

The user data unregistered SEI Message may be used in systems which are required to associate a highly accurate, microsecond resolution time stamp with the video frame. This UTC time stamp shall be derived from GPS as described in section 4 of [5] and will be formatted as defined in Annex A of [5]. The user data unregistered message consists of two fields as defined below:

Uuid_iso_iec_11578 is a 16 Byte field that shall be set to the value:

Bytes 1-8:	0x4D, 0x49, 0x53, 0x50, 0x6D, 0x69, 0x63, 0x72,
Bytes 9-16:	0x6F, 0x73, 0x65, 0x63, 0x74, 0x69, 0x6D, 0x65

This represents the ASCII string: "MISPmicrosectime"

User_data_payload_bytes is a variable length field. For this application, 12 bytes will be used as follows:

Byte 1:			Status
	Bit 7	0 =	GPS Locked (internal clock locked to GPS)
		1 =	GPS Flywheel (internal clock not locked to GPS, so it is running on an internal oscillator)
	Bit 6	0 =	Normal (time incremented normally since last message)
		1 =	Discontinuity (time has not incremented normally since last message)
	Bit 5	0 =	Forward (If Bit 6 = 1, this indicates that the time jumped forward)
		1 =	Reverse (If Bit 6 = 1, this indicates that the time jumped backwards)
	Bits 4-0:		Reserved (= 1)
Bytes 2, 3:			Two MS bytes of Time Stamp (microseconds)
Byte 4:			Start Code Emulation Prevention Byte (0xFF)
Bytes 5, 6:			Two next MS bytes of Time Stamp (microseconds)
Byte 7:			Start Code Emulation Prevention Byte (0xFF)
Bytes 8, 9:			Two next LS bytes of Time Stamp (microseconds)
Byte 10:			Start Code Emulation Prevention Byte (0xFF)
Bytes 11, 12:			Two LS bytes of Time Stamp (microseconds)

This represents the 64 bit microsecond UTC time where Byte 2=MSB, Bytes 3, 5, 6, 8, 9, 11 are intermediate bytes and Byte 12 = LSB. Byte 1 is transmitted first.

5 Time Stamping Metadata

Systems which are capable of time stamping both the video stream and the metadata stream have all of the information necessary to multiplex this information together in a synchronized motion imagery stream.

The structure for KLV metadata is defined in [13]. The KLV element "User Defined Time Stamp (microseconds since 1970)" is typically used as the time stamp in a KLV stream. The definition and format of this KLV element is defined in [5].

6 Carriage of Metadata in Transport Stream

If the requirements for a motion imagery system dictate that a metadata element is to be associated with a particular frame of video, or that the time associated with the metadata element is correlated to the same time line as the video, then [1] shall be used to transport the video and associated metadata in an MPEG-2 Transport Stream.

6.1 Asynchronous Carriage of Metadata

The transport of KLV metadata over MPEG-2 transport streams and program streams in an asynchronous manner shall be confined to the method defined in [11] Section 4.1.1 or Section 4.1.2. As shown in Figure 2, the metadata PES packets do not use Presentation Time Stamps (PTS) or Metadata Access Unit Wrappers. The relationship between the metadata and the video frames is typically established by their proximity in the video stream. This type of metadata carriage may be used to transport static metadata, or metadata which is not tied closely in time to the video.

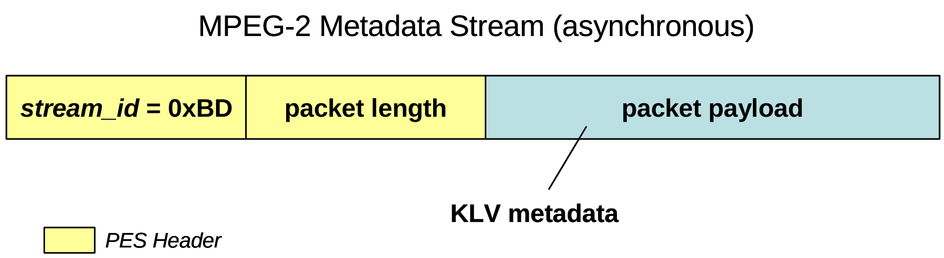

Figure 2: Asynchronous Metadata Stream

<u>Metadata PES Stream</u>

- The *stream_id* shall be 0xBD, indicating "private_stream_1."
- The *data_alignment_indicator* shall be set to one when the PES packet contains the beginning of a KLV item, and shall be set to zero otherwise.
- The delay of any data through the System Target Decoder buffers shall be less than or equal to one second. (This ensures that the metadata is in close proximity to the video frames that it relates to.)

  NOTE: Careful use of the buffer size and leak rate for metadata defined in the System Target Decoder (STD) Model (and as specified in the *metadata_std_ descriptor*) can force a closer proximity of the metadata to the associated frame of video.

- When inserting asynchronous metadata into a transport stream which already carries asynchronous metadata of the same *format_identifier*, it shall be inserted into the existing metadata elementary stream.

Program Map Table (PMT)

- The *stream_type* shall be 0x06, indicating "PES packets containing private data."

- The Metadata Stream shall be defined in the PMT as a separate Stream within the same Program as the Video Elementary Stream. [1] allows for multi-program Transport Streams, and methods for associating metadata in one program to video in another. Multi-program Transport Streams are not covered within the scope of this STANDARD.

- For legacy compliance with [11], the program element loop in the PMT shall contain a *registration_descriptor* as defined in [1], and the *format_identifier* field shall be set to 0x4B4C5641 (KLVA).

- The PMT shall contain a *metadata_descriptor* for each metadata service within the metadata stream. The *metadata_descriptor* shall be within the descriptor loop for the metadata stream. The *metadata_descriptor* contains the *metadata_service_id* for the service it describes. The following values are used to identify metadata types within the *metadata_descriptor*:

  metadata_format = 0xFF (specified by *metadata_format_identifier*)
  metadata_format_identifier = 0x4B4C5641 (KLVA)

  NOTE: Earlier versions of [1] describe the use of the *registration_descriptor* to "uniquely and unambiguously identify formats of private data." The *metadata_descriptor*, however, provides more functionality, and is therefore specified.

- The PMT shall contain a single *metadata_std_descriptor* for the metadata stream.

- The PMT may contain other descriptors such as the *content_labeling_descriptor* and the *metadata_pointer_descriptor*.

The following is a sample *registration_descriptor*, *metadata_descriptor* and *metadata_std_descriptor* for a metadata stream containing asynchronous KLV metadata:

registration_descriptor
descriptor_tag = 0x05 (5)
descriptor_length = 0x04 (4)
format_identifier = 0x4B4C5641 = "KLVA"

metadata_descriptor
descriptor_tag = 0x26 (38)
descriptor_length = 0x09 (9)

metadata_application_format = 0x0100-0x0103 (see Table 1)
metadata_format = 0xFF
metadata format identifier = 0x4B4C5641 = "KLVA"
metadata_service_id = 0x00
decoder_config_flags = '000'
DSM-CC_flag = '0'
reserved = '1111'

metadata_std_descriptor
descriptor_tag: 0x27 (39)
descriptor_length: 0x09 (9)
reserved = '11'
metadata_input_leak_rate: (determined by encoder)
reserved = '11'
metadata_buffer_size: (determined by encoder)
reserved = '11'
metadata_output_leak_rate: (determined by encoder)

Note that the *metadata_output_leak_rate* must be specified for asynchronous metadata.

metadata_application_format (type of KLV metadata)	
0x0100	General
0x0101	Geographic Metadata
0x0102	Annotation Metadata
0x0103	Still Image on Demand

Table 1: KLV metadata type

6.2 Synchronous Carriage of Metadata

Several ways to carry metadata over MPEG-2 transport streams are detailed in [1]. This STANDARD requires the use of the method outlined in Section 2.12.4 "Use of PES packets to transport metadata" for transporting metadata that is synchronized with the essence stream. This method provides a way to synchronize metadata with video using the Presentation Time Stamp (PTS) found in the Packetized Elementary Stream (PES) header. This time stamp is coded in the MPEG-2 Systems PES layer, and is relevant for H.264 as well as MPEG-2.

The metadata may or may not be sampled at the same time as a video frame depending upon the system design. If it is sampled at the same time as a video frame, the metadata and video frame will have the same PTS. If the metadata is not sampled at the same time as the video frame, it will be stamped with a different PTS, but exist on the same timeline as the video frame.

Figure 3 shows the general structure of a PES packet in the metadata bit stream. In the most common implementation, the packet payload would consist of a single metadata cell which includes a five-byte header followed by KLV metadata.

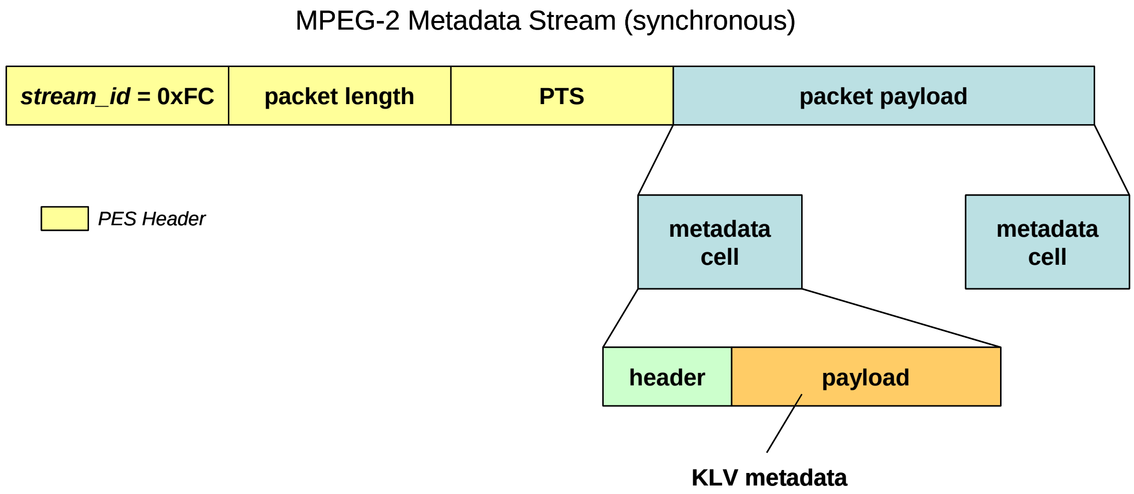

Figure 3: Synchronous Metadata Stream

A metadata service is defined in [1] as "a coherent set of metadata of the same format delivered to a receiver for a specific purpose." When transporting metadata using this service, a unique *metadata_service_id* is assigned to each service. Each metadata service is represented by a collection of metadata access units which are transported in PES packets.

Details of the implementation of this method are given below.

<u>Metadata PES Stream</u>

- Insertion of synchronous metadata requires the use of ISO 13818-1 § 2.12.4.
- The *stream_id* shall be 0xFC, indicating "metadata stream".
- Each PES packet shall have a PTS to be used to synchronize the metadata with the video frames.
- In each PES packet that carries metadata, the first PES packet data byte shall be the first byte of a Metadata Access Unit Cell.
- The PTS in the PES header shall apply to each Access Unit contained in the PES packet.
- The PTS shall signal the time that the metadata Access Unit becomes relevant. It is assumed that the metadata is decoded instantaneously (i.e., no DTS shall be coded). If a video frame and a metadata Access Unit have the same PTS, then they were sampled at the same time.

- Each metadata Access Unit may be carried in one or more Access Unit Cells.

- The delay of any data through the System Target Decoder buffers shall be less than or equal to one second. (This ensures that the metadata is in close proximity to the video frames that it relates to.)

  NOTE: Careful use of the buffer size and leak rate for metadata defined in the System Target Decoder (STD) Model (and specified in the *metadata_std_ descriptor*) can force a closer proximity of the metadata to the associated frame of video.

- When inserting synchronous metadata into a transport stream which already carries synchronous metadata new metadata shall be added to the existing synchronous metadata stream as a new service or an existing service.

  NOTE: redefinition of buffer size and/or leak rate for metadata may be necessary to maintain the desired proximity between metadata and associated video frames.

<u>Program Map Table (PMT)</u>

- The *stream_type* shall be 0x15, indicating "Metadata carried in PES packets."

- The Metadata Stream shall be defined in the PMT as a separate stream within the same Program as the Video Elementary Stream. [1] allows for multi-program Transport Streams, and methods for associating metadata in one program to video in another. Multi-program Transport Streams are not covered within the scope of this STANDARD.

- The PMT shall contain a *metadata_descriptor* for each metadata service within the metadata stream. The *metadata_descriptor* shall be within the descriptor loop for the metadata stream. The *metadata_descriptor* contains the *metadata_service_id* for the service it describes. The following values are used to identify metadata types within the *metadata_descriptor*:
  metadata_format = 0xFF (specified by metadata format identifier)
  metadata_format_identifier = 0x4B4C5641 "KLVA"

- The PMT shall contain a single *metadata_std_descriptor* for the metadata stream.

- The PMT may contain other descriptors such as the *content_labeling_descriptor* and the *metadata_pointer_descriptor*.

  The following is a sample *metadata_descriptor*, *metadata_std_descriptor* and *metadata_AU_cell* header for a metadata stream containing synchronous KLV metadata.

  metadata_descriptor
  descriptor_tag = 0x26 (38)
  descriptor_length = 0x09 (9)
  metadata_application_format = 0x0100-0x0103 (see Table 1)
  metadata_format = 0xFF

metadata format identifier = 0x4B4C5641 = “KLVA”
metadata_service_id = 0x00
decoder_config_flags = ‘000’
DSM-CC_flag = ‘0’
reserved = ‘1111’

metadata_std_descriptor
descriptor_tag: 0x27 (39)
descriptor_length: 0x09 (9)
reserved = ‘11’
metadata_input_leak_rate: (determined by encoder)
reserved = ‘11’
metadata_buffer_size: (determined by encoder)
reserved = ‘11’
metadata_output_leak_rate: (unspecified; recommend setting to 0)

Note that the *metadata_output_leak_rate* is unspecified for synchronous metadata. The recommended value is 0.

Metadata_AU_cell (5-byte header)
metadata_service_id = 0x00 *(8 bits)*
sequence_number = (supplied by encoder; increments each cell – *8 bits*)
cell_fragmentation_indication = ‘11’, ‘10’, ‘01’ or ‘00’ *(2 bits)*
decoder_config_flag = ‘0’ *(1 bit)*
random_access_indicator = ‘0’ or ‘1’ *(1 bit)*
reserved = ‘1111’ *(4 bits)*
AU_cell_data_length = (supplied by encoder) *(16 bits)*

7 Transition Annex

Many motion imagery systems have been developed based on [11] for asynchronous carriage of metadata in MPEG-2 Transport Streams. A synchronous method for transporting metadata with the associated video streams is provided in [1]. As systems advance and strive for more accurate metadata, migrating to this new method of transporting metadata is important.

New systems and applications must be capable of handling metadata using both the format in [11] and [1]. It will be relatively straightforward for motion imagery systems to add support for [1]. Minor changes must be made to the transport layer (multiplexing and demultiplexing) of the motion imagery stream.